

```

1 // [反片語/Anagrams]
2 #define IN "P0507IN.txt"
3 #define OUT "P0507OUT.txt"
4 //*****
5 #include <iostream>
6 #include <time.h>
7 using namespace std;
8 void redir(void);
9 //*****
10 /* Work Space*/
11 #include <string>
12 #include <vector>
13 #include <algorithm>
14 #include <map>
15
16 string normalize(string &s);
17 //*****
18 int main(void)
19 {
20     redir(); //redirection
21 //*****
22 /* Work Space*/
23     string s, n;
24     vector<string> words, ans;
25     map<string, int> cnt;
26     int i;
27
28     while(cin >> s){
29         if(s[0] == '#'){
30             break;
31         }
32         words.push_back(s);
33         n = normalize(s);
34
35         if(!cnt.count(n)){
36             cnt[n] = 0;
37         }
38         cnt[n]++;
39     }
40
41     for(i=0; i<words.size(); i++){
42         if(cnt[normalize(words[i])] == 1){
43             ans.push_back(words[i]);
44         }
45     }
46     sort(ans.begin(), ans.end());
47
48     for(i=0; i<ans.size(); i++){
49         cout << ans[i] << endl;
50     }
51 //*****
52     freopen("CON", "r", stdin); //取消重新導向
53     freopen("CON", "w", stdout);
54
55     printf("Time used = %.2f\n", (double)clock()/CLK_TCK); //傳回程式目前為止執行的時間
56
57     system("pause");
58     return 0; //the end...
59 }
60
61 void redir(void)
62 {
63     freopen(IN, "r", stdin);

```

```

64     freopen(OUT, "w", stdout);
65 }
66 /**
67  * Work Space*/
68 //把每個單字標準化(轉為小寫字母再按字母排序)
69 string normalize(string &s)
70 {
71     string ans = s;
72     int i;
73
74     for(i=0; i<s.length(); i++){
75         ans[i] = tolower(ans[i]);
76     }
77
78     sort(ans.begin(), ans.end());
79     return ans;
80 }
81
82 //Input(IN) Sample
83 /*
84 ladder came tape soon leader acme RIDE lone Dreis peat
85 ScAlE orb eye Rides dealer NotE derail LaCes drIed
86 noel dire Disk amce Rob Dries
87 #
88 */

```